Lawrence Ouyang
504128219
Math 151B

Homework 6

S1: Using the Householder reflection code below:

```
function [ A ] = Householder( n, A )
%
% Householder(n, A)
% Obtains symmetric diagonal matrix A(n-1) similar to the symmetric matrix
% A using Householder's method.
% n: dimension
% A: symmetric matrix

for k = 1:(n-2)
    q = 0; alpha = 0; PROD = 0;
    v = zeros(n,1); u = zeros(n,1); z = zeros(n,1);

    for j = (k+1):n
        q = q + A(j,k)^2;
    end

    if A(k+1,k) == 0
        alpha = -sqrt(q);
    else
        alpha = -(sqrt(q)*A(k+1,k)/abs(A(k+1,k)));
    end

    RSQ = alpha^2 - alpha*A(k+1,k);
    v(k+1) = A(k+1,k) - alpha;

    for j = (k+2):n
        v(j) = A(j,k);
    end

    for j = k:n
        for i = (k+1):n
            u(j) = u(j) + A(j,i)*v(i);
        end
        u(j) = u(j)/RSQ;
    end

    for i = (k+1):n
        PROD = PROD + v(i)*u(i);
    end

    for j = k:n
        z(j) = u(j) - (PROD/(2*RSQ))*v(j);
```

```
    end

    for l = (k+1):n-1
        for j = (l+1):n
            A(j,l) = A(j,l) - v(l)*z(j) - v(j)*z(l);
            A(l,j) = A(j,l);
        end
        A(l,l) = A(l,l) - 2*v(l)*z(l);
    end
    A(n,n) = A(n,n) - 2*v(n)*z(n);

    for j = (k+2):n
        A(k,j) = 0;
        A(j,k) = 0;
    end
    A(k+1,k) = A(k+1,k) - v(k+1)*z(k);
    A(k,k+1) = A(k+1,k);
end
end
```

Applying our formula to the given A results in the following symmetric tridiagonal matrix:

$$A = \begin{bmatrix} 4 & 1 & -1 & 0 \\ 1 & 3 & -1 & 0 \\ -1 & -1 & 5 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix} \rightarrow T = \begin{bmatrix} 4 & -\sqrt{2} & 0 & 0 \\ -\sqrt{2} & 5 & -\sqrt{3} & 0 \\ 0 & -\sqrt{3} & 5 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

S2: Using a simple implementation of Rayleigh Quotient Iteration below:

```
function [lambda, v] = RayleighQuotient(n, A, v, N)
%
% RayleighQuotient(n, A, v, N)
% Approximate an eigenvalue and eigenvector of A using the Rayleigh
% Quotient Iteration.
% n: dimension of A
% A: matrix
% v: approximate eigenvector x, norm(x, inf) = 1
% N: maximum number of iterations
% lambda: approximate eigenvalue

lambda = (v.'*A*v)/(v.'*v);
for k = 1:N
    w = (A - lambda*eye(n))\v;
    v = w/norm(w, inf);
    lambda = (v.'*A*v)/(v.'*v);
end
end
```

This gives us

$$\lambda = 2.4859, v = \begin{bmatrix} -1.5156 \\ 1.6038 \\ -0.6910 \\ 0.9127 \end{bmatrix}$$

S3: Using the simple implementation of Simulataneous Iteration below:

```
function [lambda, W] = SimultaneousIteration(A, V, N)
%
% SimultaneousIteration(A, V, N)
% Approximate an eigenvalue and eigenvector of A using the Simultaneous
% Iteration method.
% A: matrix
% V: matrix of orthogonal column entries
% N: maximum number of iterations
% lambda: approximate eigenvalue vector
% W: matrix with eigenvector columns

[Q,R] = qr(V);

for k = 1:N
    W = A*Q;
    [Q,R] = qr(W);
end
lambda = diag(Q.'*A*Q);
end
```

The two largest eigenvalues are:

$$\lambda_1 = 7.0861, v_1 = \begin{bmatrix} -2.3562 \\ -1.8929 \\ 5.3786 \\ 3.4856 \end{bmatrix}$$

$$\lambda_2 = 4.4280, v_2 = \begin{bmatrix} -3.1861 \\ -1.8739 \\ -0.5102 \\ -2.3841 \end{bmatrix}$$